|  |
| --- |
| **PROYECTO: Manual tecnico** |
| **201313965– Luis Fernando Gonzalez Real**  **201807200 – Jhonatan Josue Paredes Ochoa**  **201602458 – María Luisa Fernanda Calderón Molina** |

**Resumen**

La empresa Desktop Records, decide que es momento de presentar su información de una manera más cómoda para los usuarios, por lo que se desea implementar un sitio web capaz de procesar, manipular y presentar la información de la empresa.

Este programa permitirá al usuario gestionar los catálogos de empleados, discos y países de una manera más cómoda y eficiente permitiendo un manejo más detallado de los datos.

**Palabras clave**

**Programa:** Proyecto o planificación ordenada de las distintas partes o actividades que componen algo que se va a realizar

**Sitio web:** Es un conjunto de páginas web que se encuentran relacionadas por el contenido

**Base de datos:** Es un conjunto de datos estructurados que pertenecen a un mismo contexto

**Cliente:** Es una aplicación informática o un ordenador que consume un servicio remoto en otro ordenador conocido como servidor

**Servidor:** Es un sistema que proporciona recursos, datos, servicios o programas a otros ordenadores, conocidos como clientes

***Abstract***

*The Desktop Records Company decides that it is time to present its information in a more comfortable way for users, so it wants to implement a website capable of processing, manipulating and presenting the company's information.*

*This program will allow the user to manage the catalogs of employees, albums and countries in a more comfortable and efficient way, achieving a more detailed management of the data..*

***Keywords***

***Program:*** *Project or orderly planning of the different parts or activities that make up something that is going to be carried out*

***Website:*** *It is a set of web pages that are related by the content*

***Database:*** *It is a set of structured data that belong to the same context*

***Client:*** *It is a computer application or a computer that consumes a remote service on another computer known as a server.*

***Server:*** *It is a system that provides resources, data, services or programs to other computers, known as clients.*

**Introducción**

Se describen los diferentes funcionamientos del programa informático, basándonos en un enfoque técnico.

Así como también las herramientas utilizadas para la elaboración del programa y la especificación de cada módulo tanto del backend como del frontend

**Desarrollo del tema**

**Django** (framework utilizado para el frontend)

Es un framework de desarrollo web de código abierto, escrito en Python, que respeta el patrón de diseño conocido como modelo–vista–controlador (MVC). Fue desarrollado originalmente para gestionar páginas web orientadas a noticias de la World Company de Lawrence.

**Flask** (framework utilizado para el backend)

Es un framework minimalista escrito en Python que permite crear aplicaciones web rápidamente y con un mínimo número de líneas de código. Está basado en la especificación WSGI de Werkzeug y el motor de templates Jinja2 y tiene una licencia BSD.

**Python**

Es un lenguaje de alto nivel de programación interpretado cuya filosofía hace hincapié en la legibilidad de su código.

Se trata de un lenguaje de programación multiparadigma, ya que soporta parcialmente la orientación a objetos, programación imperativa y, en menor medida programación funcional. Es un lenguaje interpretado, dinámico y multiplataforma.

Administrado por Python Software Foundation, posee una licencia de código abierto, denominada Python Software Foundation License.​ Python se clasifica constantemente como uno de los lenguajes de programación más populares.

**XML**

Por sus siglas en inglés de eXtensible Markup Language, traducido como 'Lenguaje de Marcado Extensible', es un metalenguaje que permite definir lenguajes de marcas desarrollado por el World Wide Web Consortium (W3C) utilizado para almacenar datos en forma legible. Proviene del lenguaje SGML y permite definir la gramática de lenguajes específicos para estructurar documentos grandes. A diferencia de otros lenguajes, XML da soporte a bases de datos, siendo útil cuando varias aplicaciones deben comunicarse entre sí o integrar información.

**Graphviz** (Graph Visualization)

Es un conjunto de herramientas de software para el diseño de diagramas definido en el lenguaje descriptivo DOT.​ Fue desarrollado por AT&T Labs​ y liberado como software libre con licencie tipo Eclipse.

**Backend**

***GET - /empleados***

Devolverá la información de todos los empleados de cada departamento.

***POST - /empleadoNombre***

Se enviará un nombre y la petición devolverá la información de los empleados que contengan el nombre indicado.

***POST - /empleadoDepartamento***

Se enviará un departamento y la petición devolverá la información de los empleados que pertenezcan al departamento indicado.

***POST - /empleadoSueldo***

Se enviará un sueldo y la petición devolverá la información de los empleados que contengan el sueldo indicado.

***POST - /agregarEmpleado***

Se enviará una estructura JSON con la información del empleado (departamento, ID, nombre, puesto, salario) y la petición ejecutará código para poder ingresar el empleado al departamento.

***POST - /modificarEmpleado***

Se enviará una estructura JSON con la información del empleado (ID, nombre, puesto, salario) y la petición ejecutará código para poder modificar el empleado con el ID enviado.

***POST - /eliminarEmpleado***

Se enviará una estructura JSON con la información del empleado (ID) y la petición ejecutará código para poder eliminar el empleado del departamento.

***GET - /reporteEmpleados***

Esta petición ejecutará código para generar el gráfico de Graphviz con la información de los empleados.

***GET - /discos***

Devolverá la información de todos los discos.

***POST - /discoTitulo***

Se enviará un nombre y la petición devolverá la información de los discos que contengan el nombre indicado.

***POST - /discoYear***

Se enviará un año y la petición devolverá la información de todos los discos que pertenezcan al año indicado.

***POST - /discoArtista***

Se enviará un nombre y la petición devolverá la información de todos los discos que contengan el nombre indicado.

***POST - /agregarDisco***

Se enviará una estructura JSON con la información del disco (título, artista, país, compañía, precio y año) y la petición ejecutará código para poder ingresar el disco al catálogo.

***POST - /modificarDisco***

Se enviará una estructura JSON con la información del disco (título, artista, país, compañía, precio y año) y la petición ejecutará código para poder modificar el disco con el título indicado.

***POST - /eliminarDisco***

Se enviará una estructura JSON con la información del disco (título) y la petición ejecutará código para poder eliminar el disco del departamento.

***GET - /reporteDiscos***

Esta petición ejecutará código para generar el gráfico de Graphviz con la información de los discos del catálogo.

***GET - /paises***

Devolverá la información de todos los países de cada continente.

***POST - /paisMoneda***

Se enviará un tipo de moneda y la petición devolverá la información de los países que contengan la moneda indicada..

***POST - /paisIdioma***

Se enviará un idioma y la petición devolverá la información de todos los países que contengan el idioma indicado.

***POST - /contienente***

Se enviará un nombre de continente y la petición devolverá la información de todos los países de ese continente.

***POST - /agregarPais***

Se enviará una estructura JSON con la información del país (continente, nombre, moneda, capital, idioma, población) y la petición ejecutará código para poder ingresarlo.

***POST - /modificarPais***

Se enviará una estructura JSON con la información del país (continente, nombre, moneda, capital, idioma, población) y la petición ejecutará código para poder modificarlo.

***GET - /reporteRegiones***

Esta petición ejecutará código para generar el gráfico de Graphviz con la información de los países registrados en la plataforma.

**Frontend**

**Ver empleados**

Se utiliza la petición GET /empleados.

**Buscar según nombre**

Se utiliza la petición POST /empleadoNombre

**Buscar según departamento**

Se utiliza la petición POST /empleadoDepartamento.

**Buscar según sueldo**

Se utiliza la petición POST /empleadoSueldo

**Agregar empleado**

Se enviará la información en formato JSON para poder usar la petición POST /agregarEmpleado.

**Modificar empleado**

Se enviará la información en formato JSON para poder usar la petición POST /modificarEmpleado

**Eliminar empleado**

Se enviará la información en formato JSON para poder usar la petición POST /eliminarEmpleado

**Reporte empleados**

Se debe utilizar la petición GET /reporteEmpleados. Se debe de mostrar la imagen en el frontend.

**Ver discos**

Se utiliza la petición GET /discos.

**Buscar discos según nombre**

Se utiliza la petición POST /discoTitulo

**Buscar discos según año**

Se utiliza la petición POST /discoYear.

**Buscar discos según artista**

Se utiliza la petición POST /discoArtista

**Agregar disco**

Se enviará la información en formato JSON para poder usar la petición POST /agregarDisco

**Modificar disco**

Se enviará la información en formato JSON para poder usar la petición POST /modificarDisco

**Eliminar disco**

Se enviará la información en formato JSON para poder usar la petición POST /eliminarDisco

**Reporte discos**

Se debe utilizar la petición GET /reporteDiscos. Se debe de mostrar la imagen en el frontend.

**Ver regiones**

Se utiliza la petición GET /países.

**Buscar países según moneda**

Se utiliza la petición POST /paisMoneda

**Buscar países según idioma**

Se utiliza la petición POST /paisIdioma.

**Buscar países según continente**

Se utiliza la petición POST /continente

**Agregar país**

Se enviará la información en formato JSON para poder usar la petición POST /agregarPais

**Reporte regiones**

Se debe utilizar la petición GET /reporteRegiones. Se debe de mostrar la imagen en el frontend.

**Reportes**

Para los reportes, se deben generar gráficos utilizando Graphviz, con todos los nodos y nodos hijos, que permita ver la estructura completa del archivo.

**Conclusiones**

Se creó un servicio web amigable con el usuario.

El programa informático está a la vanguardia de la tecnología.

El manejo de seguridad de la información es confiable.

**Referencias bibliográficas**

Máximo 5 referencias en orden alfabético.

C. J. Date, (1991). *An introduction to Database Systems.* Addison-Wesley Publishing Company, Inc.